Mathematics and Society
University of Isfahan
http://math-sci.ui.ac.ir

Journal of Mathematics and Society
ISSN (print): 2345-6493, ISSN (on-line): 2345-6507
Vol. 10 No. 4 (2025), pp. 1–24.

http://www.ui.ac.ir

BAYESIAN NEURAL NETWORKS; WHY AND HOW?

MOEIN MONEMI, S. MAHMOUD TAHERI, S. MORTEZA AMINI*

Abstract. One of the main challenges in utilizing neural networks is the problem of overfitting. This occurs when a neural network model fits the training data too precisely, but fails to generalize to data outside the training set. This lack of generalization is often observed when the number of training samples is smaller than the number of features being analyzed, and the complexity of the model — that is, the number of weights and biases in the neural network — is high. In such situations, ensemble learning, and more specifically bagging methods, are commonly employed. These methods use resampling techniques to incorporate uncertainty into the model, thereby improving the model's ability to generalize. However, when the training sample size is extremely limited, resampling becomes less effective, and the uncertainty introduced in the model is very limited. Bayesian neural networks address this by quantifying parameter uncertainty and considering parameter states that may not have been observed in the existing data. This leads to a significant improvement in model generalization. In addition to mitigating overfitting, this approach also provides access to the posterior predictive distribution, allowing for the calculation of prediction intervals. In this article, we briefly review Bayesian neural networks, explain how they are trained, and then analyze data and compare these models with standard neural networks.

Keywords: Artificial neural networks, Bayesian inference, Posterior distribution, MCMC, Multiple linear regression, Classification
Article Type: Promotional Paper.
Communicated by Afshin Parvardeh.
Received: 03-06-2024, Accepted: 14-10-2024, Published Online: 29-06-2025.
Cite this article: M. Monemi, S. M. Taheri, S. M. Amini, Bayesian Neural Networks; why and how?, *Journal of Mathematics and Society*, **10** no. 4 (2025) 1–24.
http://dx.doi.org/10.22108/msci.2024.141722.1668 .

1. Introduction

Bayesian Neural Networks (BNNs) are extensively used in statistical machine learning due to their ability to use prior knowledge about network parameters, such as weights and biases. One of the main challenges in traditional neural networks is the problem of overfitting, which occurs when the model fits the training data too precisely, thereby failing to generalize well to unseen test data. BNNs, by quantifying uncertainty, allow for consideration of parameter states that may not have been seen in the training data, thereby improving model robustness. In addition to reducing overfitting, BNNs employ the posterior predictive distribution to predict new data and derive prediction intervals, which offer a measure of uncertainty in the predictions. Nevertheless, obtaining the posterior distribution poses significant challenges due to the intractable integrals in the denominator of Bayes' theorem—an issue particularly pertinent in models like BNNs, which rely on activation functions, as illustrated in Figures 2 and 3. To overcome this, approximate inference methods such as Markov Chain Monte Carlo (MCMC), Variational Bayes, and Expectation Propagation are utilized to approximate the posterior distribution.

In this paper, we review Bayesian Neural Networks that utilize a multilayer perceptron, similar to those depicted in Figures 1 and 4, and detail the process of training them using the MCMC method, specifically employing the Metropolis-Hastings algorithm [8]. This method generates a sequence of dependent random samples, that typically exhibit a first-order Markov property. In first-order Markov processes, the conditional distribution of any state, given all previous states, depends only on the previous state. One of the most well-known methods for achieving this, is the Metropolis-Hastings algorithm introduced by [8], which uses a proposal distribution to generate samples from the unknown distribution such as the posterior distribution, which we will implement for both regression and classification tasks.

In the regression task, we assume we have the inputs variable $X \in \mathbb{R}^{n\times d}$ and the outputs variable $y \in \mathbb{R}^{n\times p}$, where n is the number of inputs, and d and p represent the dimensionality of the input and output variables, respectively. Our goal is to model $\hat{y} = f(X) + \epsilon$, with $\epsilon \sim \mathcal{N}_n(0, \sigma^2 I_n)$, requiring us to approximate the posterior distribution $p(w, b, \sigma^2|X, y)$. Here, w and b represent the weights and biases of the neural network, and σ denotes the standard deviation of the likelihood function $p(y|X, W, b, \sigma^2)$, where $y|X, W, b, \sigma^2 \sim \mathcal{N}_n(f(X), \sigma^2 I_n)$. The Metropolis-Hastings algorithm in this task uses proposal distributions for the posterior parameters (w, b, σ^2) and generates samples accordingly. After T iterations, a density estimator is used to approximate the posterior distribution.

In the classification task, similar to regression, we have input variables $X \in \mathbb{R}^{n\times d}$ and output variables $y \in \mathbb{R}^{n\times k}$, where k represents the number of classes. Each row of y is a basis vector, with only one element equal to 1 and the rest equal to 0. The objective in this problem is to classify

http://dx.doi.org/10.22108/msci.2024.141722.1668

the inputs X based on the given data. For this task, we aim to estimate the posterior distribution $p(w, b|X, y)$.

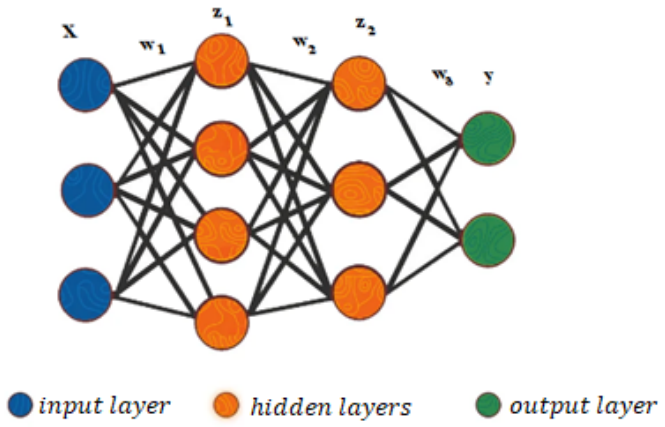


FIGURE 1. The architecture of two hidden layers Neural Network

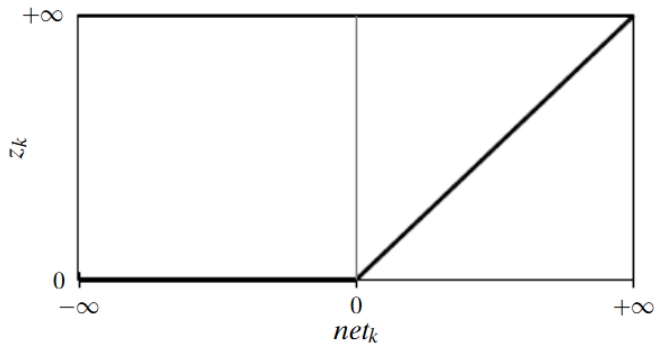


FIGURE 2. Relu activation function [18]

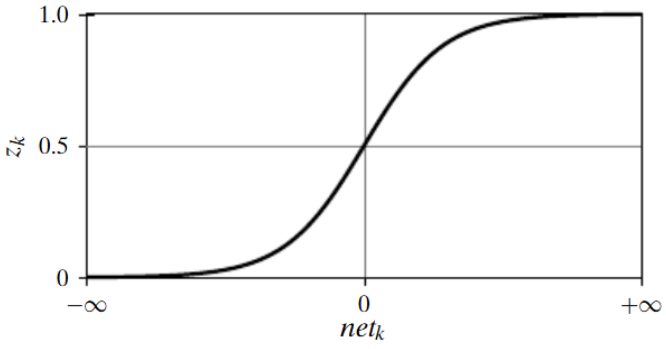


FIGURE 3. Sigmoid activation function [18]

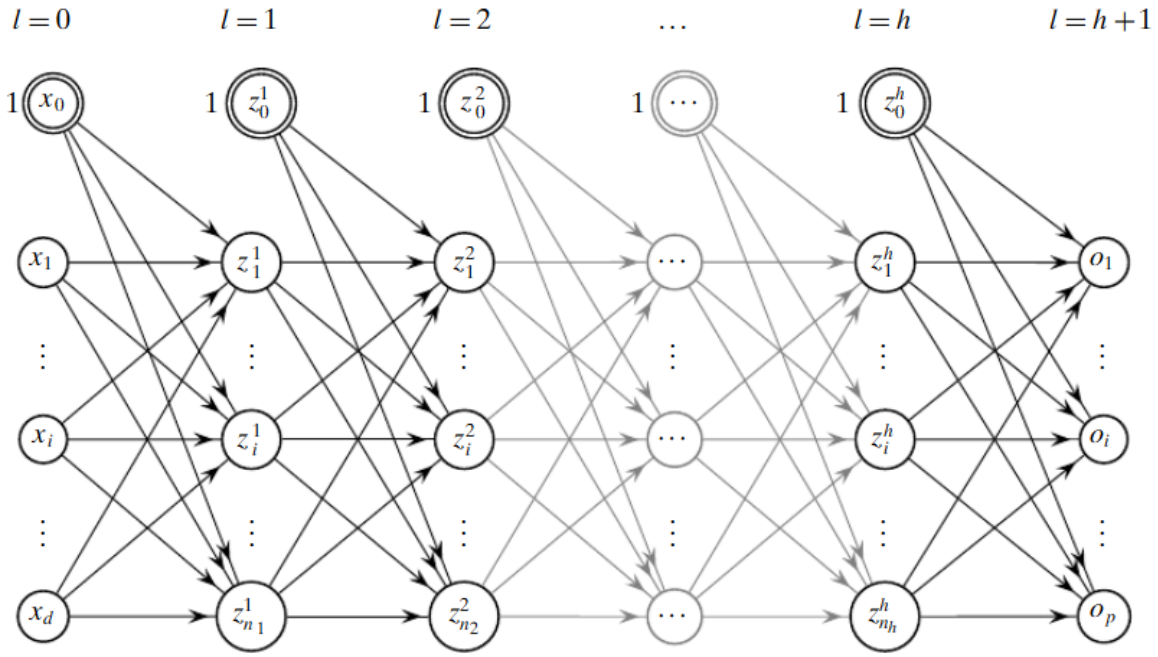


FIGURE 4. The architecture of the Neural Network consists of $h+2$ layers, where h represents the number of hidden layers, with the input and output layers completing the structure [18]

2. Main Results

In this section, we evaluate the performance of BNNs against standard neural networks on several datasets. The results demonstrate that BNNs outperform traditional neural networks in both regression and classification tasks. Figures 5 and 6 respectively illustrate the residuals from the regression task and the ROC curve from the classification task. Tables 1 and 2 compare the performance of the two approaches across various evaluation metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), accuracy, F-measure, and others.

	BNN	**ANN**
Mean Square Error	1.21	30.75
Mean Absolute Error	0.88	4.80

TABLE 1. Comparison of the performance of BNN and standard NN on the Riboflavin dataset.

http://dx.doi.org/10.22108/msci.2024.141722.1668

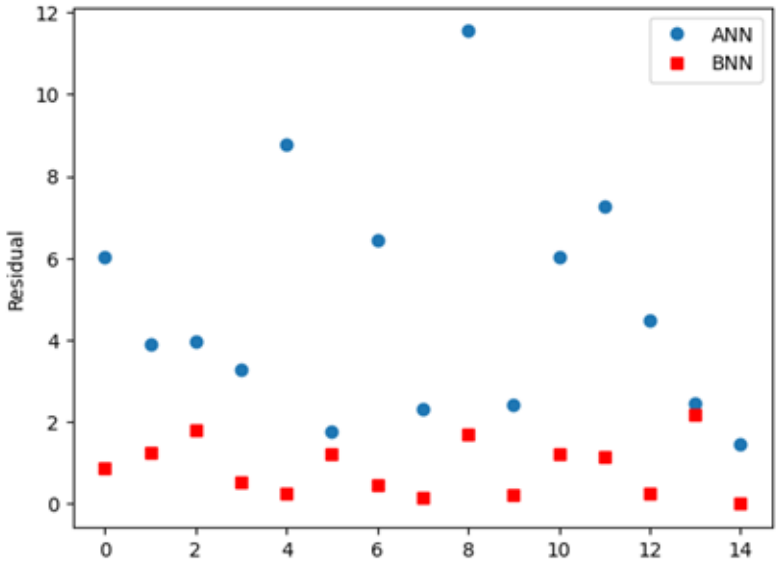


FIGURE 5. Residual values for 15 data points in the regression task on the Riboflavin dataset, comparing BNN (represented by square markers) and standard NN (represented by circle markers)

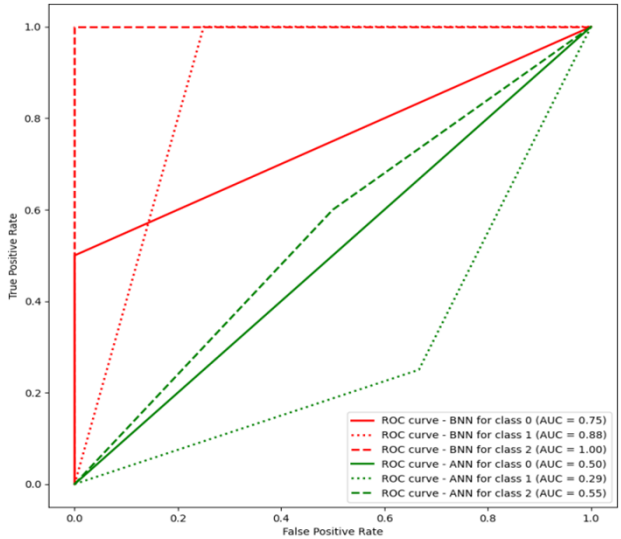


FIGURE 6. ROC curve of BNN versus standard NN in classification task on lung cancer dataset (The continuous curve: for class 0, dotted curve: for class 1, dashed curve: for class 2)

	BNN	ANN
Accuracy	0.85	0.42
F-measure	0.84	0.38
F-Beta measure	0.86	0.36
Jaccard measure	0.75	0.27

TABLE 2. Evaluation of BNN and NN on the lung cancer dataset based on four common metrics.

3. Conclusions

Bayesian neural networks was reviewed as an efficient alternative to classical neural networks. In two practical examples, it was observed that under certain conditions, Bayesian neural networks perform significantly better than classical neural networks. Since there is no conjugate prior and posterior distribution for neural networks based on the activation functions used, the Metropolis-Hastings sampling method was employed to approximate the posterior distribution. It should be noted that, this method incurs high computational costs when dealing with high-dimensional parameters. Therefore, recent research has shifted toward using more cost-effective posterior approximation methods, such as Variational Bayes, [1], [12], [16], and [18]. Future work could explore faster approximation methods (such as Variational Bayes) and improvements in sampling methods like $MCMC$ for Bayesian neural networks.

Moein Monemi
School of Engineering Science, College of Engineering, University of Tehran Tehran, Iran
Email: moein.monemi@ut.ac.ir

S. Mahmoud Taheri
School of Engineering Science, College of Engineering, University of Tehran Tehran, Iran
Email: sm_taheri@ut.ac.ir

S. Morteza Amini
Department of Statistics, School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran Tehran, Iran
Email: morteza.amini@ut.ac.ir

http://dx.doi.org/10.22108/msci.2024.141722.1668

ریاضی و جامعه
شاپا (چاپی): ۲۳۴۵-۶۴۹۳، شاپا(الکترونیکی): ۲۳۴۵-۶۵۰۷
جلد ۱۰، شماره ۴ (۱۴۰۴)، صص. ۱-۲۴

http://www.ui.ac.ir
http://math-sci.ui.ac.ir

شبکه‌های عصبی بیزی؛ چرایی و چگونگی

معین منعمی، سید محمود طاهری، سید مرتضی امینی*

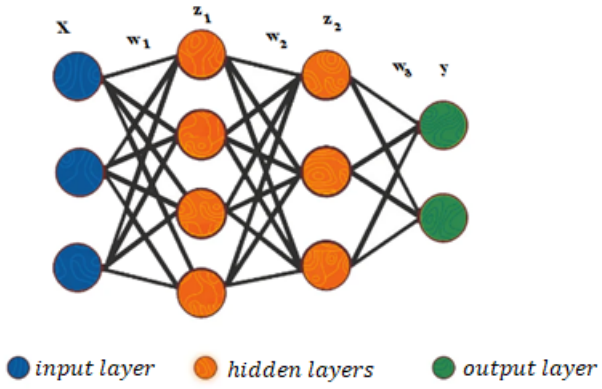
چکیده. یکی از چالش‌ها در به‌کارگیری شبکه‌های عصبی، مشکل بیش‌برازش[1] است. این مشکل زمانی پیش می‌آید که مدل شبکه عصبی به داده‌های آموزشی به‌طور دقیق برازش داده می‌شود، ولی این مدل به داده‌های خارج از این مجموعه قابل تعمیم نیست. عدم تعمیم‌پذیری مدل بیشتر در شرایطی پیش می‌آید که تعداد نمونه‌های مجموعه داده آموزشی کم‌تر از تعداد ویژگی‌های مورد بررسی و پیچیدگی مدل یعنی تعداد وزن‌ها و اریبی‌های شبکه عصبی است. در چنین وضعیتی معمولاً از یادگیری ترکیبی[2] و به‌طور خاص از روش‌های دسته‌بندی[3] استفاده می‌شود. در این روش از بازنمونه‌گیری برای ایجاد عدم قطعیت در مدل استفاده می‌شود و به‌دین وسیله تعمیم‌پذیری مدل بهبود پیدا می‌کند. با این‌حال بازنمونه‌گیری در شرایطی که اندازه نمونه آموزشی بسیار کم است، کارایی ندارد و عدم قطعیت ایجاد شده در مدل بسیار محدود است. شبکه‌های عصبی بیزی با کمّی‌سازی عدم قطعیت پارامترها، حالاتی از پارامترها را در نظر می‌گیرند که ممکن است توسط داده‌های موجود دیده نشده باشند. بدین‌ترتیب تعمیم‌پذیری مدل افزایش چشم‌گیر پیدا می‌کند. این روش علاوه بر جلوگیری از بیش‌برازش، توزیع پیش‌بین پسین را نیز در اختیار ما قرار می‌دهد و امکان به‌دست آوردن بازه‌های پیش‌بینی را نیز فراهم می‌آورد. در این مقاله به معرفی شبکه‌های عصبی بیزی و نحوه آموزش آن‌ها و سپس به تحلیل داده‌ها و مقایسه این مدل‌ها با شبکه‌های عصبی عادی می‌پردازیم.

۱. درآمد

یادگیری عمیق باعث ایجاد تحولی در یادگیری ماشین شده است و راه‌حل‌هایی برای برخی مسائل ارائه می‌دهد که حل آن‌ها به روش‌های تحلیلی و سنتی دشوار است. با وجود این، مدل‌های یادگیری عمیق، با مشکلاتی از قبیل بیش‌برازش روبه‌رو هستند و نمی‌توان آن‌ها را برای حالت‌های دیده نشده، تعمیم داد [۱۶]. در شبکه‌های عصبی عمیق از روش پس انتشار[1] برای برآورد پارامترها استفاده می‌شود. این عمل، در مسائلی که خطاهای کوچک ممکن است به تغییرات چشم‌گیر منجر شود، مانند تشخیص پزشکی مشکل‌ساز است [۱۳]. به‌منظور جلوگیری از این مشکل چندین رویکرد بر پایهٔ کمی‌سازی عدم قطعیت ارائه شده‌اند که این مشکلات را تا حدودی برطرف می‌کنند [۱۲]. یادگیری ترکیبی و به‌طور خاص روش‌های دسته‌بندی با استفاده از بازنمونه‌گیری سعی در ایجاد عدم قطعیت در مدل دارند تا بدین وسیله تعمیم‌پذیری مدل، بهبود پیدا کند. با این‌حال بازنمونه‌گیری در شرایطی که اندازه نمونه آموزشی بسیار کم است، کارایی چندانی ندارد و عدم قطعیت ایجاد شده در مدل بسیار محدود است. در رویکرد بیزی این کار با استفاده از مدل‌بندی توزیعی عدم قطعیت پارامترهای مدل انجام می‌شود.

عبارات و کلمات کلیدی: شبکه‌های عصبی مصنوعی، استنباط بیزی، توزیع پسین، رگرسیون خطی چندگانه، طبقه‌بندی.
نوع مقاله: ترویجی
دبیرتخصصی رابط: افشین پرورده
تاریخ دریافت: ۱۴۰۳/۰۳/۱۴ تاریخ پذیرش: ۱۴۰۳/۰۷/۲۳ تاریخ انتشار آنلاین: ۱۴۰۴/۰۴/۰۸
ارجاع به مقاله: م. منعمی، س. م. طاهری و س. م. امینی، شبکه‌های عصبی بیزی؛ چرایی و چگونگی، نشریه ریاضی و جامعه، ۱۰ شماره ۴ (۱۴۰۴) ۱-۲۴.
http://dx.doi.org/10.22108/msci.2024.141722.1668 .

[1] back propagation



شکل ۱. نمونه‌ای از معماری یک شبکه عصبی چندلایه

Figure 1: The architecture of two hidden layers Neural Network

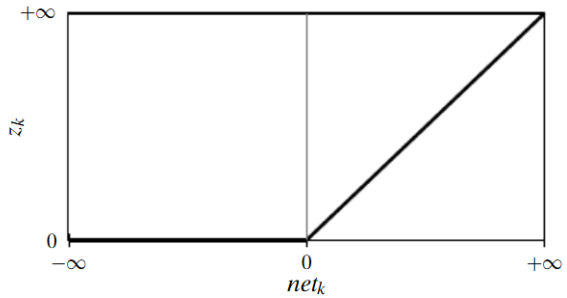
در این مقاله و در بخش ۲، مروری کوتاه بر شبکه‌های عصبی مصنوعی و در بخش ۳، مروری بر استنباط بیزی و چالش‌های آن خواهیم داشت. در بخش ۴، شبکه‌های عصبی بیزی را معرفی خواهیم کرد و معیارهای ارزیابی در بخش ۵ بررسی خواهند شد. در بخش ۶، یکی از الگوریتم‌های معروف مبتنی بر نمونه‌گیری را بررسی کرده و در بخش ۷، نحوه پیاده‌سازی این الگوریتم در دو مسئله رگرسیون و طبقه‌بندی[۲] بر پایهٔ شبکه‌های عصبی بیزی بررسی می‌شود. در بخش ۸، عملکرد شبکه‌های عصبی بیزی با عملکرد شبکه‌های عصبی مصنوعی مقایسه می‌شود و سرانجام در بخش ۹، جمع‌بندی و نتیجه‌گیری ارائه خواهد شد.

۲. شبکه‌های عصبی مصنوعی

شبکه‌های عصبی مصنوعی[۳]، یا کوتاهی شبکه‌های عصبی، الهام گرفته شده از شبکه‌های عصبی بیولوژیکی است. یک نورون بیولوژیکی یا یک سلول عصبی، شامل دندریت‌ها[۴]، یک جسم سلولی[۵] و آکسون[۶] است که به پایه‌های سیناپس[۷] می‌رود. هر نورون، اطلاعات را از طریق سیگنال‌های الکتروشیمیایی منتقل می‌کند. هنگامی‌که غلظت کافی یون در دندریت یک نورون وجود داشته باشد، یک پالس الکتریکی در امتداد آکسون خود ایجاد می‌کند که پتانسیل عمل، نامیده می‌شود و پایانه‌های سیناپس را فعال می‌کند، یون‌های بیشتری آزاد می‌کند و باعث می‌شود که اطلاعات به دندریت‌های سایر نورون‌ها جریان یابد [۱]. شبکه‌های عصبی مصنوعی را می‌توان با یک گراف جهت‌دار و وزن‌دار $G = (V, E)$ مدل‌بندی کرد. این گراف در حالت ساده، شامل یک بخش ورودی، یک بخش پنهان و یک بخش خروجی است، که هر بخش را یک لایه از شبکه می‌نامیم. تعداد لایه‌های پنهان براساس پیچیدگی مسئله، ممکن است بیشتر از یک باشد. هر گره در شبکه‌های عصبی، به‌عنوان یک لایه‌ی پردازش عمل می‌کند که ابتدا مجموع وزنی سیگنال‌های دریافتی را محاسبه و سپس توسط یک تابع غیرخطی که به آن تابع فعال‌ساز می‌گویند، خروجی جدید را تولید می‌کند و به نورون‌های لایه‌ی بعد انتقال می‌دهد. شکل ۱ نمونه‌ای از معماری یک شبکه عصبی را نشان می‌دهد [۱۹].

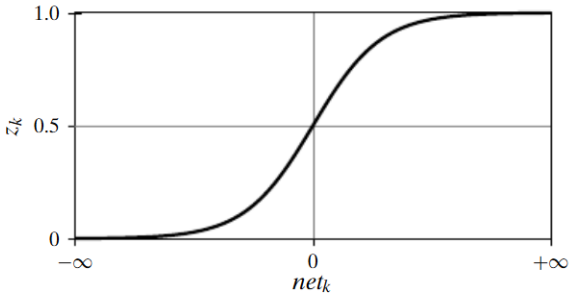
توابع فعال‌ساز بخش مهمی از هر شبکه عصبی هستند که بر روی ورودی‌های یک نورون اعمال می‌شوند. معمولاً از توابع غیرخطی به‌عنوان تابع فعال‌ساز استفاده می‌شود زیرا اکثر مدل‌ها در واقعیت، غیرخطی هستند. دو تابع فعال‌ساز رایج به شرح زیر هستند [۱۹].

[2]classification [3]artificial neural network [4]dendrites [5]cell body [6]axon [7]synapse



شکل ۲. نمودار تابع فعال‌ساز Relu

Figure 2: Relu activation function



شکل ۳. نمودار تابع فعال‌ساز سیگموئید

Figure 3: Sigmoid activation function

تابع فعال‌ساز واحد خطی یکسو شده[8]: خروجی این‌تابع، برای ورودی‌های کمتر از صفر غیرفعال و برای ورودی‌های بزرگتر از صفر، به‌صورت خطی افزایش می‌یابد. این تابع به‌عنوان خروجی یک نورون، به‌صورت زیر تعریف می‌شود:

$$z_k = \sigma(h_k) = \begin{cases} ۰, & h_k \leq ۰ \\ h_k, & h_k > ۰ \end{cases}$$

h_k ورودی نورون kام است که از مجموع وزنی نورون‌های لایه قبل به‌دست می‌آید، σ تابع فعال‌ساز و z_k خروجی نورون kام است (شکل ۲).

تابع فعال‌ساز سیگموئید[9]: خروجی این تابع، عددی بین صفر و یک است و به‌عنوان خروجی یک نورون به‌صورت زیر تعریف می‌شود (شکل ۴).

$$z_k = \sigma(h_k) = \frac{۱}{۱ + e^{-h_k}}$$

همان‌طور که اشاره شد، پارامترهای مدل شبکه عصبی، وزن‌ها و اریبی‌ها هستند که به‌ترتیب، با w و b نشان می‌دهیم. با ورودی $x \in \mathbb{R}^p$، یک شبکه عصبی، مقداری را به خروجی $y \in Y$، با استفاده از پارامترهای w و b نسبت می‌دهد. میزان خطای این مقدار از مقدار واقعی y توسط یک تابع زیان $L(y, x; w, b)$ اندازه‌گیری می‌شود. در مسئله طبقه‌بندی، هر y احتمال یک طبقه[10] است و در مسئله رگرسیون، y یک متغیر پیوسته خروجی است. در مسئله‌های رگرسیون معمولاً تابع زیان

[8]rectified linear unit (ReLU) [9]Sigmoid [10]class

http://dx.doi.org/10.22108/msci.2024.141722.1668

میانگین توان دوم خطا استفاده می‌شود و در مسئله‌های طبقه‌بندی معمولاً تابع خطای آنتروپی متقاطع[11] به‌کار برده می‌شود. در شبکه‌های عصبی سنتی، وزن‌های شبکه توسط روش انتشار معکوس و با کمینه‌سازی تابع زیان به‌صورت زیر آموزش می‌بینند [۳].

$$(w,b)_{BP} = \arg\min_{(w,b)} L(y,x;w,b)$$

دستیابی به این مقدار از w و b، توسط الگوریتم‌های بهینه‌سازی مبتنی بر گرادیان صورت می‌گیرد.

۳. آشنایی با استنباط بیزی

استنباط بیزی یکی از روش‌های پرکاربرد در استنباط آماری و به‌طور خاص در یادگیری ماشین است. در این روش، از دانش پیشین دربارهٔ پارامترها و یا مدل مجهول استفاده می‌شود. با استفاده از قانون احتمال شرطی و قانون احتمال کل، قضیه بیز به‌دست می‌آید که توسط آن می‌توان به توزیع پسین پارامترهایی رسید که مؤلفه‌های اصلی در مدل مرتبط هستند.

فرض کنیم X، متغیر تصادفی یا بردار تصادفی با تابع چگالی $f_X(x|\theta)$ است. قرار دهید $L(\theta|x) = f_X(x|\theta)$ به‌طوری‌که $L(\theta|x)$ را تابع درست‌نمایی گویند که برحسب متغیر تصادفی X، یک تابع احتمال است اما برحسب θ، لزوماً یک تابع احتمال نیست. همچنین فرض کنیم $\pi(\theta)$ تابع پیشین است که بیانگر اطلاعات پیشین دربارهٔ پارامتر θ است. $\pi(\theta)$ می‌تواند یک تابع احتمال و یا می‌تواند یک تابع ناسره[12] مانند تابع ثابت، ۱ باشد یعنی $\pi(\theta) = ۱$ که لزوماً یک تابع احتمال نیست.

در حالتی‌که پارامتر θ پیوسته است (و معمولاً چنین است)، قضیه بیز به‌صورت زیر فرمول‌بندی می‌شود:

$$f_{\theta|X}(\theta|x) = \frac{f_X(x|\theta)\cdot\pi(\theta)}{\int f_X(x|\theta)\cdot\pi(\theta)\,d\theta} = \frac{L(\theta|x)\cdot\pi(\theta)}{\int L(\theta|x)\cdot\pi(\theta)\,d\theta} \qquad (۱)$$

مخرج رابطه ۴ را شواهد[13] گویند و $f_{\theta|X}(\theta|x)$ را تابع چگالی احتمال پسین گویند.

همان‌طور که در ابتدا اشاره شد، دستیابی به توزیع پسین دقیق، گاه با چالش‌هایی روبه‌رو است، زیرا در بسیاری از حالات با انتگرال‌هایی مهارنشدنی در مخرج رابطه ۴ روبه‌رو هستیم که جواب تحلیلی برای آن‌ها وجود ندارد. یکی از روش‌های رایج در رفع این چالش، استفاده از پیشین‌های مزدوج[14] است که در ادامه، دربارهٔ آن توضیح می‌دهیم.

تعریف ۱.۳. [۸] فرض کنیم F رده‌ای از توزیع‌های نمونه‌ای $p(x|\theta)$ و P رده‌ای از توزیع‌های پیشین برای پارامتر θ باشد. گوییم رده P برای F مزدوج است اگر

$$\forall p(\cdot|\theta)\in F,\ p(\cdot)\in P \Rightarrow p(\theta|x)\in P$$

مثال ۲.۳. نمونه‌های مستقل و هم‌توزیع $X_۱, X_۲, \ldots, X_n \overset{iid}{\sim} Ber(\theta)$ را در نظر بگیرید یعنی

$$L(\theta|x_۱,x_۲,\ldots,x_n) = f_{(X_۱,\ldots,X_n)}(x_۱,x_۲,\ldots,x_n|\theta)$$

$$= \prod_{i=۱}^{n}\theta^{x_i}(۱-\theta)^{(۱-x_i)} = \theta^{\sum_{i=۱}^n x_i}(۱-\theta)^{(n-\sum_{i=۱}^n x_i)}$$

تابع بالا، همان تابع درست‌نمایی θ است که می‌توان با استفاده از روش بیشینه درست‌نمایی، مقدار θ را برآورد کرد و داریم $\hat{\theta}_{MLE} = \frac{۱}{n}\sum_{i=۱}^n x_i$. ما قصد داریم یک توزیع پیشین مزدوج برای پارامتر θ در نظر بگیریم. توجه شود که پارامتر θ در

[11] cross-entropy [12] Improper [13] evidence [14] conjugate

این مسئله احتمال موفقیت را بیان می‌کند و تمام مقادیر آن در بازه‌ی $[۰،۱]$ قرار دارد. توزیع پیشین مزدوج برای θ، توزیع بتا به‌صورت زیر است:

$$\theta \sim Beta(\alpha, \beta), \quad \pi(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-۱}(۱-\theta)^{\beta-۱}$$

بنابراین توزیع پسین پارامتر θ به‌صورت زیر صورت‌بندی می‌شود:

$$\begin{aligned} f_{\theta|X_۱,X_۲,\ldots,X_n}(\theta|x_۱,x_۲,\ldots,x_n) &= \frac{L(\theta|x_۱,x_۲,\ldots,x_n).\pi(\theta)}{\int L(\theta|x_۱,x_۲,\ldots,x_n).\pi(\theta)\ d\theta} \\ &= \frac{\theta^{\alpha-۱+\sum_{i=۱}^n x_i}(۱-\theta)^{\beta-۱+n-\sum_{i=۱}^n x_i}}{\int \theta^{\alpha-۱+\sum_{i=۱}^n x_i}(۱-\theta)^{\beta-۱+n-\sum_{i=۱}^n x_i} d\theta}, \end{aligned}$$

رابطه بالا را می‌توان به تابع چگالی بتا تبدیل کرد، کافی است صورت و مخرج رابطه بالا را در ضریب نرمال‌ساز تابع چگالی بتا ضرب کنیم. داریم

$$\begin{aligned} &f_{(\theta|X_۱,X_۲,\ldots,X_n)}(\theta|x_۱,x_۲,\ldots,x_n) \\ &= \frac{\frac{\Gamma(\alpha+\beta+n)}{\Gamma(\alpha+\sum_{i=۱}^n x_i)\Gamma(\beta+n-\sum_{i=۱}^n x_i)}\theta^{\alpha-۱+\sum_{i=۱}^n x_i}(۱-\theta)^{\beta-۱+n-\sum_{i=۱}^n x_i}}{\int \frac{\Gamma(\alpha+\beta+n)}{\Gamma(\alpha+\sum_{i=۱}^n x_i)\Gamma(\beta+n-\sum_{i=۱}^n x_i)}\theta^{\alpha-۱+\sum_{i=۱}^n x_i}(۱-\theta)^{\beta-۱+n-\sum_{i=۱}^n x_i}\, d\theta}. \end{aligned}$$

مخرج کسر بالا همان تابع چگالی بتا است که با انتگرال گیری برروی تمام مقادیر $\theta \in [۰،۱]$، مقدار آن برابر ۱ خواهد شد. بنابراین داریم:

$$f_{\theta|X_۱,\ldots,X_n}(\theta|x_۱,x_۲,\ldots,x_n) = \frac{\Gamma(\alpha+\beta+n)}{\Gamma\left(\alpha+\sum_{i=۱}^n x_i\right)\Gamma\left(\beta+n-\sum_{i=۱}^n x_i\right)}\theta^{\alpha-۱+\sum_{i=۱}^n x_i}(۱-\theta)^{\beta-۱+n-\sum_{i=۱}^n x_i}$$

و می‌نویسیم

$$\theta \mid X_۱ = x_۱, X_۲ = x_۲, \ldots, X_n = x_n \sim Beta\left(\alpha + \sum_{i=۱}^n x_i,\ \beta + n - \sum_{i=۱}^n x_i\right).$$

چون هر دو توزیع پسین و پیشین دارای توزیع بتا هستند بنابراین توزیع پیشین بتا، یک توزیع پیشین مزدوج برای θ در مدل دو جمله‌ای است.

مشاهده می‌شود که با استفاده از پیشین‌های مزدوج، جواب تحلیلی انتگرال‌های مخرج رابطه ۴ به‌راحتی به‌دست می‌آید. همچنین در این‌حالت قادریم تا خواص توزیع پسین مانند امیدریاضی، واریانس، توزیع پیش‌بین پسین و مواردی از این قبیل را محاسبه کنیم. به ادامه‌ی مثال ۲.۳ توجه کنید.

مثال ۲.۳ (ادامه). فرض کنیم X^* متغیر تصادفی جدید باشد و ما قصد داریم توزیع پیش‌بین پسین برای این متغیر تصادفی را به‌دست آوریم. داریم

$$\begin{aligned} &f_{X^*|X_۱,\ldots,X_n}(x^*=۱|x_۱,x_۲,\ldots,x_n) \\ &= \int f_{X^*}(x^*=۱|\theta) f_{\theta|X_۱,\ldots,X_n}(\theta|x_۱,x_۲,\ldots,x_n)\, d\theta = \int \theta \cdot f_{\theta|X_۱,\ldots,X_n}(\theta|x_۱,x_۲,\ldots,x_n)\, d\theta \\ &= \mathbb{E}_{\theta|X_۱,X_۲,\ldots,X_n}[\theta] = \frac{\alpha+\sum_{i=۱}^n x_i}{\alpha+\beta+n}. \end{aligned}$$

http://dx.doi.org/10.22108/msci.2024.141722.1668

رابطه بالا همان امیدریاضی توزیع پسین (یعنی توزیع بتا) است. همچنین می‌توان نقطه‌ای را برآورد کرد که تابع چگالی پسین را بیشینه کند یعنی $\hat{\theta}_{MAP} = \frac{\alpha-1+\sum_{i=1}^{n} x_i}{\alpha+\beta+n-2}$. ولی در عمل و در اکثر مسائل، نمی‌توان از پیشین‌های مزدوج استفاده کرد. در این حالات با انتگرال‌هایی مهار نشدنی روبه‌رو هستیم. به مثال زیر توجه کنید.

مثال ۳.۳. فرض کنیم در مسئله طبقه‌بندی به روش رگرسیون لجستیک، متغیرهای مستقل $x_1, x_2, \ldots, x_n$ متغیرهایی تک‌بعدی باشند. متغیرهای پاسخ در این مسئله به‌صورت زیر تعریف می‌شوند:

$$y_i|x_i, w_0, w_1 \sim Ber\left(\frac{e^{(w_0+w_1 x_i)}}{1+e^{(w_0+w_1 x_i)}}\right)$$

که در رابطه بالا $w_0 + w_1 x_i$ همان مدل رگرسیون خطی ساده است که در تابع سیگموئید مورد استفاده قرار می‌گیرد. تابع درست‌نمایی در این مسئله به‌صورت زیر است:

$$\begin{aligned} f(y_1, \ldots, y_n, x_1, \ldots, x_n|w_0, w_1) &= \prod_{i=1}^{n} f(y_i|x_i, w_0, w_1) \\ &= \prod_{i=1}^{n} \left(\frac{e^{w_0+w_1 x_i}}{1+e^{w_0+w_1 x_i}}\right)^{y_i} \cdot \left(\frac{1}{1+e^{w_0+w_1 x_i}}\right)^{1-y_i} \\ &= e^{\left(w_0 \sum_{i=1}^{n} y_i\right)+\left(w_1 \sum_{i=1}^{n} x_i y_i\right)-\sum_{i=1}^{n} \log(1+e^{w_0+w_1 x_i})}. \end{aligned}$$

در این مسئله هدف، دستیابی به توزیع پسین $f(w_0, w_1|y_1, \ldots, y_n, x_1, \ldots, x_n)$ است. در این‌حالت توزیع پیشینی برای پارامترهای مدل یعنی w_0, w_1 وجود ندارد که مزدوج باشد و معمولاً از توزیع‌های پیشین ناآگاهی‌بخش[15] در این مسائل استفاده می‌کنیم. به‌عنوان مثال می‌توان از توزیع نرمال با میانگین ۰ و واریانس بسیار بزرگ برای پیشین استفاده کرد. بنابراین نمی‌توان به‌طور دقیق این مسئله را حل کرد و این یک چالش بزرگ در روش‌های بیزی است که موجب ظهور روش‌های تقریب توزیع پسین می‌شود. همچنین در مدل‌های پیچیده‌تر مانند شبکه‌های عصبی بیزی، با چنین چالشی روبه‌رو هستیم.

اکنون که با مفاهیم مقدماتی شبکه‌های عصبی و استنباط بیزی آشنا شدیم، به موضوع اصلی این مقاله، یعنی شبکه‌های عصبی بیزی می‌پردازیم.

۴. شبکه‌های عصبی بیزی

شبکه‌های عصبی بیزی، شبکه‌هایی هستند که با قرار دادن توزیع پیشین بر روی پارامترهای شبکه شامل وزن‌ها و اریبی‌ها، حالت‌هایی از پارامترها را در نظر می‌گیرند که ممکن است توسط داده‌های موجود دیده نشده باشند. این عمل، باعث تعمیم‌پذیری مدل می‌شود. همچنین، جلوگیری از مشکل بیش‌برازش در بسیاری از مجموعه داده‌ها یکی از مزایای شبکه‌های عصبی بیزی است.

در شبکه‌های عصبی بیزی، هدف اصلی دستیابی به توزیع پسین پارامترهای شبکه است که چنین صورت‌بندی می‌شود [۳]

$$p(w, b|x, y) = \frac{p(y|x, w, b)\, p(w, b)}{p(x, y)}.$$

[15] non informative

اگر به برآورد نقطه‌ای مناسب نیاز داشته باشیم، کافی است مقداری از w و b را انتخاب کنیم که توزیع پسین را بیشینه می‌کند. چنین برآوردی را برآورد بیشینه احتمال پسین[۱۶] گویند و به‌صورت زیر تعریف می‌شود:

$$\begin{aligned}(w,b)_{MAP} &= \arg\max_{w,b} \log\left(p\left(w,b|x,y\right)\right)\\ &= \arg\max_{w,b} \left(\log\left(p\left(y|x,w,b\right)\right)+\log\left(p\left(w,b\right)\right)\right).\end{aligned}$$

یکی از برتری‌های رویکرد بیز، به‌دست آوردن تابع چگالی برای خروجی‌های یک ورودی جدید است که به آن توزیع پیش‌بین پسین می‌گویند و برپایهٔ رابطه‌ی زیر محاسبه می‌شود:

$$f\left(y^{\text{new}}|x^{\text{new}},x,y\right)=\int p\left(y^{\text{new}}|x^{\text{new}},w,b\right)\, p\left(w,b|x,y\right)\, d(w,b),$$

که x^{new} ورودی جدید و y^{new} خروجی متناسب با این ورودی است. رابطه‌ی بالا را همچنین می‌توان به‌صورت زیر بیان کرد:

$$f\left(y^{\text{new}}|x^{\text{new}},x,y\right)=\mathbb{E}_{p(w,b|x,y)}\left[p\left(y^{\text{new}}|x^{\text{new}},w,b\right)\right] \qquad (۲)$$

این رابطه بیانگر استفاده از توزیعی از شبکه‌های عصبی برروی داده‌ی جدید به‌جای استفاده از تنها یک شبکه عصبی است.

مزایای شبکه‌های عصبی بیزی نسبت به شبکه‌های عصبی سنتی عبارت‌اند از [۱۱] :

- کمی‌سازی عدم قطعیت: BNNها روشی اصولی برای کمی‌سازی عدم قطعیت در یادگیری عمیق ارائه می‌دهند و با این کار، حالت‌هایی از وزن‌ها و اریبی‌ها را در نظر می‌گیرند که ممکن است توسط داده‌های موجود دیده نشده باشد.
- استواری[۱۷] نسبت به بیش‌برازش: BNNها می‌توانند بیش‌برازش را با ترکیب مدل‌های مختلف روی پارامترهای شبکه کاهش دهند که به تعمیم‌پذیری بهتر مدل کمک می‌کند.
- مدیریت بهتر مجموعه داده‌های کوچک: BNNها می‌توانند در زمانی که داده‌های موجود محدود هستند، کاراتر از شبکه‌های عصبی عادی باشند، زیرا می‌توانند عدم‌قطعیت مرتبط با داده‌های محدود را به‌طور مؤثرتری دریافت کنند.
- منظم‌سازی[۱۸]: استفاده از توزیع‌های پیشین در BNNها می‌تواند به‌عنوان شکلی از منظم‌سازی عمل کند یعنی زمانی‌که از برآورد نقطه‌ای برای برآورد بهترین پارامتر استفاده کنیم، تابع دیگری به تابع هدف اضافه می‌شود که برای منظم‌سازی استفاده می‌شود. بیشتر روش‌های منظم‌سازی در شبکه‌های عصبی را که از برآورد نقطه‌ای استفاده می‌کنند را می‌توان با دیدگاه بیزی و استفاده از یک پیشین مناسب بیان کرد.

به‌تازگی فناوری‌های بسیاری بر اساس این الگو در زمینه‌های مختلف از جمله شناسایی تصویر[۱۹] توسعه داده شده‌اند مانند شناسایی کشتی از تصویر رادار، تشخیص پزشکی، عیب‌یابی دستگاه‌های پیچیده و سیستم پشتیبانی تصمیم‌گیری حیاتی [۴].

پیشتر گفتیم که یکی از مسائل مهم در استنباط بیزی، به‌دست آوردن ضریب نرمال‌سازی (مخرج رابطه) است که عموماً راه‌حل تحلیلی برای حل آن وجود ندارد. دراین‌باره از روش‌های تقریب توزیع پسین استفاده می‌شود که متداول‌ترین آن‌ها روش‌های نمونه‌گیری مانند زنجیره مارکوف مونت کارلو[۲۰] هستند که در بخش ۶ شرح داده می‌شوند.

[16] maximum a posteriori probability [17] robustness [18] regularization [19] vision recognition [20] Markov Chain Monte Carlo (MCMC)

http://dx.doi.org/10.22108/msci.2024.141722.1668

۵. معیارهای ارزیابی

در این بخش به بررسی برخی معیارهای ارزیابی در مسئله رگرسیون و طبقه‌بندی می‌پردازیم که در این پژوهش مورد استفاده قرار گرفته‌اند.

۱) در صورتی‌که خروجی شبکه عصبی یک متغیر پیوسته باشد (مانند مدل‌سازی رگرسیونی)، معمولاً از دو معیار ارزیابی خطا، یعنی میانگین توان دوم خطا و میانگین قدر مطلق خطا استفاده می‌شود. به‌ازای مقدار پیش‌بینی شده $\hat{y}_i$ از متغیر پاسخ y_i به‌ازای $i = ۱, \dots, n$، این دو معیار چنین محاسبه می‌شوند:

$$MSE = \frac{۱}{n}\sum_{i=۱}^{n}(\hat{y}_i - y_i)^۲, \qquad MAE = \frac{۱}{n}\sum_{i=۱}^{n}|\hat{y}_i - y_i|$$

۲) در مسئله طبقه‌بندی، زمانی‌که متغیر خروجی y_i، نشان دهنده‌ی طبقه داده iام و $\hat{y}_i$ طبقه پیش‌بینی شده‌ی داده iام را نشان می‌دهد، یک معیار متداول برای ارزیابی، دقت[۲۱] طبقه‌بندی است که به‌صورت زیر تعریف می‌شود:

$$A = \frac{۱}{n}\sum_{i=۱}^{n} I(\hat{y}_i = y_i)$$

که در آن، $I(\hat{y}_i = y_i)$ تابع نشانگر[۲۲] درستی پیش‌بینی است که مقدار آن برابر ۱ است اگر $\hat{y}_i = y_i$ وگرنه برابر ۰ است.

۳) معیار متداول دیگر $F-$اندازه[۲۳] است که برابر است با

$$F = \frac{۱}{k}\sum_{i=۱}^{k} F_i, \qquad F_i = \frac{۲}{\frac{۱}{R_i} + \frac{۱}{A_i}}, \quad i = ۱, \dots, k$$

که در آن، A_i نسبت تعداد پیش‌بینی‌های درست به تعداد کل داده‌های طبقه i است، یعنی $A_i = \frac{n_{ii}}{m_i}$ که n_{ii} تعداد داده‌هایی است که به درستی در طبقه i پیش‌بینی شده‌اند و m_i تعداد کل داده‌های پیش‌بینی شده در طبقه i است، و $R_i = \frac{n_{ii}}{n_i}$ که n_i تعداد داده‌های واقعی در طبقه i را نشان می‌دهد.

۴) (نمره F-Beta[۲۴]). این نمره تعمیم داده شده معیار $F-$اندازه است که شامل یک پارامتر تنظیم β است. زمانی‌که $\beta = ۱$، این معیار دقیقاً همان معیار $F-$اندازه خواهد بود. به‌ازای مقادیر $\beta > ۱$، این معیار وزن بیشتری را به R و وزن کمتری را به A اختصاص می‌دهد، و برعکس. این معیار زمانی مورد استفاده قرار می‌گیرد که تعداد داده‌هایی که به‌درستی در طبقه i پیش‌بینی شده‌اند (R_i)، دارای اهمیت بیشتری نسبت به دقت طبقه i (A_i) باشد. این معیار به‌ازای هر طبقه i، با رابطه زیر محاسبه می‌شود:

$$F_{i_\beta} = (۱ + \beta^۲)\frac{A_i R_i}{(\beta^۲ A_i + R_i)}$$

۵) (نمره یاکارد[۲۵]). معیار متداول دیگر به‌منظور ارزیابی طبقه‌بندی، نمره یاکاراد است که چنین تعریف می‌شود

$$J(y,\ \hat{y}) = \frac{|\hat{y}\ \cap\ y|}{|\hat{y}| + |y| - |\hat{y}\ \cap y|}$$

که در آن، $\hat{y}$ خروجی‌های پیش‌بینی توسط طبقه‌بند[۲۶] و y خروجی‌های واقعی و $|A|$ تعداد اعضای مجموعه A است.

۶) (منحنی مشخصه عامل گیرنده[۲۷]). رسم و بررسی این منحنی یک روش رایج در ارزیابی عملکرد طبقه‌بند دوکلاسه است که البته می‌تواند برای طبقه‌بندی چندکلاسه نیز استفاده شود. اگر این منحنی را برای طبقه i رسم کنیم، محور x در این منحنی

[21] accuracy [22] indicator [23] F-measure [24] F-Beta score [25] jaccard score [26] classifier [27] receiver operating characteristic (ROC)

نرخ داده‌هایی را نشان می‌دهد که به اشتباه در طبقه i پیش‌بینی شده و محور y این منحنی نرخ داده‌هایی را نشان می‌دهد که به‌درستی در این طبقه پیش‌بینی شده‌اند. مساحت زیر این منحنی را به اختصار AUC[۲۸] می‌نامند. هرچه این مقدار برای طبقه i به ۱ نزدیک‌تر باشد، بیانگر بهتر بودن طبقه‌بند برای این طبقه است.

۶. روش‌های نمونه‌گیری از توزیع پسین

پیشتر گفتیم که دستیابی به توزیع پسین دقیق، در بسیاری از مسائل، شدنی نیست و این یک چالش در استنباط بیزی است. برای رفع این چالش از روش‌های تقریب توزیع پسین استفاده می‌شود. یکی از روش‌های معروف، نمونه‌گیری از توزیع پسین است. در ادامه، الگوریتم متروپولیس هستینگز[۲۹] که یکی از روش‌های نمونه‌گیری است را بررسی می‌کنیم.

عملکرد این روش به گونه‌ای است که دنباله‌ای از نمونه‌های تصادفی وابسته که عموماً دارای ویژگی مارکوف مرتبه اول هستند، از توزیع پسین تولید می‌شود. در ویژگی مارکوف مرتبه اول گوییم، توزیع شرطی یک نمونه به شرط تمام نمونه‌های قبلی، برابر با توزیع شرطی نمونه به شرط فقط نمونه قبلی آن است. یکی از معروف‌ترین این روش‌ها الگوریتم متروپولیس هستینگز است که در ادامه به معرفی آن می‌پردازیم.

۱.۶. الگوریتم متروپولیس هستینگز. الگوریتم متروپولیس–هستینگز در سال ۱۹۷۰ توسط هستینگز ارائه شد [۱۰]. این الگوریتم، تعمیم یافته الگوریتم متروپولیس است که توسط متروپولیس و اولام در سال ۱۹۴۹ و پس از آن توسط متروپولیس و همکاران در سال ۱۹۵۳ ارائه شد [۹] و [۱۰]. الگوریتم۱ به‌طور کلی یک دنباله از نمونه‌ها به‌صورت زیر از تابع چگالی $p(\theta\,|D)$ تولید می‌کند.

الگوریتم ۱ الگوریتم متروپولیس هستینگز

ورودی: تابع چگالی $p(\theta\,|D)$.
خروجی: یک دنباله از نمونه‌ها.

شروع

۱: با یک مقدار اولیه $\theta^{(0)}$ شروع کنید
۲: قرار دهید $t=1$.
۳: قرار دهید $\theta^{t}=\theta^{(t-1)}$.
۴: نمونه پیشنهادی را از توزیع پیشنهاده $q(\theta'|\theta^{t})$ تولید کنید.
۵: قرار دهید $\alpha=\min\left(1,\frac{P(D|\theta')p(\theta')q(\theta^{t}|\theta')}{P(D|\theta^{t})p(\theta^{t})q(\theta'|\theta^{t})}\right)$.
۶: با احتمال α قرار بدهید $\theta^{(t)}=\theta'$ وگرنه قرار بدهید $\theta^{(t)}=\theta^{(t-1)}$.
۷: قرار بدهید $t=t+1$.
۸: تا زمانی که $t\leq T$ به مرحله ۳ بروید وگرنه به مرحله ۹ بروید.
۹: پایان.

تابع $q(\theta'|\theta^{t})$ یک توزیع پیشنهاده است که حول پارامتر قبلی، پارامتر جدید را پیشنهاد می‌کند و با یک احتمال پذیرش، پارامتر پیشنهادی، پذیرفته و یا رد می‌شود. در انتهای الگوریتم، یک نمونه T تایی داریم اما همه‌ی این نمونه‌ها کاربردی نیستند و ما باید بخشی از ابتدای نمونه‌ها را حذف کنیم زیرا نمونه‌های ابتدایی، معمولاً نمونه‌های خوبی نیستند. ازطرفی ممکن است بین

[28] area under the ROC curve [29] Metropolis Hastings

http://dx.doi.org/10.22108/msci.2024.141722.1668

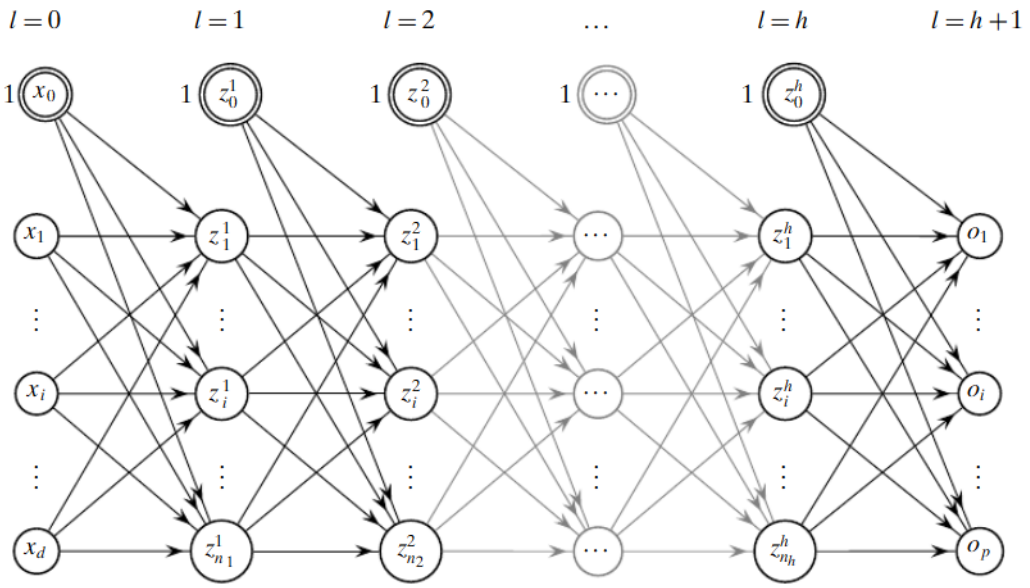
این نمونه‌ها خودهمبستگی وجود داشته باشد و نمونه‌های منتخب باید به‌گونه‌ای از نمونه اصلی انتخاب شوند که خودهمبستگی کمتری داشته باشند که برای این‌کار می‌توان از تابع خودهمبستگی[۳۰] استفاده کرد [۸].

۷. نمونه‌گیری از توزیع پسین در شبکه‌های عصبی بیزی

در این بخش روش نمونه‌گیری متروپولیس هستینگز از توزیع پسین پارامترها را برای حل مسئله رگرسیون و طبقه‌بندی بر اساس شبکه عصبی پرسپترون چند لایه بیان می‌کنیم [۵].

۱.۷. شبکه‌های عصبی بیزی برای مسئله رگرسیون. فرض کنیم $X = (X_1, \ldots, X_n)$ داده‌های ورودی و $y = (y_1, y_2, \ldots, y_n)$ بردار مقادیر متغیر پاسخ پیوسته است که $X_i \in \mathbb{R}^d$ و $y_i \in \mathbb{R}^p$. یک شبکه‌ی عصبی با لایه‌های $l_0, \ldots, l_{h+1}$ را در نظر بگیرید که l_0 لایه ورودی و l_{h+1} لایه‌ی خروجی است. همچنین فرض می‌کنیم که لایه‌ی ورودی دارای d نورون و لایه‌ی خروجی دارای p نورون است.
ورودی هر نورون i در لایه‌ی j را با $h_i^{(j)}$ و خروجی آن را با $z_i^{(j)}$ نمایش می‌دهیم. همچنین تعداد نورون لایه‌ی j را با n_j نمایش می‌دهیم و برای $j = 1, \ldots, h$ داریم.

$$h_i^{(j)} = \sum_{r=1}^{n_{j-1}} w_{ri}.z_i^{(j-1)} + b_{j-1}, \qquad z_i^{(j)} = \sigma(h_i^{(j)})$$



شکل ۴. معماری شبکه عصبی با $h + ۲$ لایه شامل h لایه پنهان [۱۹]

Figure 4: The architecture of the Neural Network consists of h + 2 layers, where h represents the number of hidden layers, with the input and output layers completing the structure [19]

[30] autocorrelation function (ACF)

در رابطه بالا، w_{ri} وزن یال از نورون r لایه‌ی قبل به نورون i لایه‌ی بعد است. همچنین b_{j-1} پارامتر اریبی در شبکه عصبی است که به لایه‌ی j اعمال می‌شود. تابع $\sigma(\cdot)$ تابع فعال‌ساز در شبکه عصبی است. برای لایه‌ی اول

$$z_i^{(0)} = X_{gi}, \qquad g = 1, \ldots, n; i = 1, \ldots, d$$

که X_{gi} داده‌ی g–ام است که مقدار i–ام آن در نورون i–ام قرار دارد. در استفاده از شبکه عصبی برای رگرسیون قرار می‌دهیم

$$o_i = z_i^{(h+1)} = h_i^{(h+1)}, \qquad i = 1, \ldots, p$$

پارامترهای شبکه عصبی بیزی را با $\theta = (W, b)$ و σ^2 نشان می‌دهیم که W وزن‌های موجود در لایه‌ها و b برداری است که اریبی در هر لایه را نشان می‌دهد. هدف یافتن $p(\theta, \sigma^2|X, y)$ یعنی توزیع پسین پارامترها است. بر پایهٔ داده‌های ورودی X، تابع درست‌نمایی خروجی‌های y به‌صورت زیر است:

$$p\left(y|X, \theta, \sigma^2\right) = \prod_{j=1}^{n} p\left(y_j|X_j, \theta, \sigma^2\right)$$

در رابطه‌ی بالا معمولاً قرار می‌دهیم $p\left(y_j|X_j, \theta, \sigma^2\right) = N_p\left(y_j; z_j^{(h+1)}, \sigma^2 I_p\right)$ یعنی فرض می‌کنیم که خروجی‌ها دارای توزیع نرمال p متغیره با بردار میانگین $z_j^{(h+1)}$ و ماتریس کواریانس $\sigma^2 I_p$ هستند. در رویکرد بیزی به رگرسیون خطی چندگانه، توزیع پیشین برای θ معمولاً به‌صورت زیر منظور می‌شود:

$$\theta \sim N(0, \tau_0^2 I_{|\theta|})$$

که در آن، $|\theta|$ نشان‌دهنده‌ی تعداد کل پارامترهاست که شامل اریبی‌ها و وزن‌های شبکه می‌باشد. همچنین برای واریانس متغیر پاسخ که مستقل از θ در نظر گرفته می‌شود، از یک توزیع پیشین گامای معکوس[۳۱] به‌صورت زیر استفاده می‌شود:

$$\sigma^2 \sim INV\_Gamma(\alpha_0, \beta_0)$$

که پارامترهای $\tau_0^2,\ \alpha_0, \beta_0$ ثابت و قابل تنظیم هستند و کاربر آن‌ها را انتخاب می‌کند برای مثال، درحالتی‌که اطلاعات پیشین دربارهٔ پارامترها موجود نباشد، از توزیع‌های پیشین ناآگاهی‌بخش استفاده می‌شود. توزیع پسین پارامترها به‌صورت زیر محاسبه می‌شود:

$$p(\theta, \sigma^2|X, y) \propto p(y|X, \theta, \sigma^2)\, p(\theta)\, p(\sigma^2)$$

باتوجه به اینکه معمولاً بر پایهٔ توزیع‌های پیشین، دستیابی به توزیع پسین دشوار است، روش رایج تقریب توزیع پسین به روش نمونه‌برداری متروپولیس–هستینگز است. درابتدا یک توزیع پیشنهادی برای پارامترها در نظر می‌گیریم. ما از توزیع نرمال برای پارامترها به‌صورت زیر استفاده می‌کنیم:

$$\theta_p \sim N_{|\theta|}\left(\theta_c,\ c_s^2 I_{|\theta|}\right)$$

$$\log \sigma_p^2 \sim N\left(\sigma_c^2, d_s^2\right)$$

که θ_p پارامترهای پیشنهادی برای وزن‌ها و اریبی‌ها و σ_p^2 پارامتر پیشنهادی برای واریانس خطا هستند. θ_c وزن‌ها و اریبی‌ها و σ_c^2 واریانس نمونه‌های فعلی هستند که به‌عنوان میانگین توزیع نرمال پیشنهادی قرار می‌گیرند. c_s^2 و σ_c^2 پارامترهای قابل تنظیم و بیانگر واریانس توزیع پیشنهادی هستند. بزرگ بودن مقدار واریانس، نرخ پذیرش را کاهش و کوچک بودن آن نرخ پذیرش را افزایش می‌دهد. نرخ پذیرش بیانگر درصد نمونه‌های پذیرفته شده در الگوریتم متروپولیس است. رابرت و همکاران

[31] inverse gamma

http://dx.doi.org/10.22108/msci.2024.141722.1668

[۱۵]، این نرخ را در بازه [۰/۲۳, ۰/۴۵] در نظر گرفته‌اند که نرخ ۰/۲۳ برای ابعاد بسیار بالا و ۰/۴۵ برای حالت تک‌بعدی است. الگوریتم متروپلیس–هستینگز برای شبکه‌های عصبی با خروجی پیوسته به‌صورت زیر است:

الگوریتم ۲ الگوریتم متروپولیس هستینگز برای شبکه‌های عصبی با خروجی پیوسته

ورودی: .

خروجی: .

شروع

۱: با مقادیر اولیه θ_0 و σ_0^2 شروع کنید.

۲: قرار دهید $t = 1$.

۳: قرار دهید $\theta_t = \theta_{(t-1)}$ و $\sigma_t^2 = \sigma_{(t-1)}^2$.

۴: نمونه‌های پیشنهادی را از توزیع $\theta_p \sim N_{|\theta|}\left(\theta_t, c_s^2 I_{|\theta|}\right)$ و $\log \sigma_p^2 \sim N\left(\sigma_t^2, d_s^2\right)$ تولید کنید.

۵: قرار دهید $\alpha = \min\left(1, \frac{P(D|\theta')p(\theta')q(\theta^t|\theta')}{P(D|\theta^t)p(\theta^t)q(\theta'|\theta^t)}\right)$.

۶: با احتمال α قرار بدهید $\theta_t = \theta_p$ و $\sigma_t^2 = \sigma_p^2$ و با احتمال $1-\alpha$ قرار بدهید $\theta_t = \theta_{(t-1)}$ و $\sigma_t^2 = \sigma_{(t-1)}^2$.

۷: قرار دهید $t = t + 1$.

۸: تا زمانی که $t \leq T$ به ۳ بروید وگرنه به مرحله ۹ بروید.

۹: پایان.

۲.۷. توزیع پیش‌بین پسین در مسئله رگرسیون. فرض کنید x^{new} ورودی جدید در مسئله رگرسیون است. بعد از تولید نمونه‌ها و دستیابی به توزیع پسین تقریبی پارامترهای شبکه، هدف پیش‌بینی مقدار متغیر خروجی y^{new} به‌ازای x^{new} است. قبلاً بیان شد که این کار توسط توزیع پیش‌بین پسین در رابطه انجام می‌شود. با این‌حال، امید ریاضی رابطه به‌دلیل در دسترس نبودن توزیع پسین، به‌طور دقیق حل‌پذیر نیست. در روش نمونه‌گیری از توزیع پسین، از میانگین نمونه استفاده می‌شود. برای این‌کار در ابتدا از تابع درست‌نمایی $p(y^{new}|x^{new}, \theta_i, \sigma_i^2)$، m نمونه تولید می‌کنیم که σ_i^2 واریانس نمونه iام و θ_i وزن‌ها و اریبی‌های نمونه iام در الگوریتم متروپلیس–هستینگز هستند. سپس مقدار خروجی پیش‌بینی شده را به‌صورت زیر محاسبه می‌کنیم.

$$\hat{y}^{new} = \frac{1}{k}\sum_{i=1}^{k}\left(\frac{1}{m}\sum_{j=1}^{m}\hat{y}_j^i\right)$$

که $\hat{y}_j^i$ نمونه تولید شده jام توسط نمونه پارامتر iام را نشان می‌دهد.

۳.۷. شبکه‌های عصبی بیزی برای مسئله طبقه‌بندی. شبکه عصبی برای طبقه‌بندی مشابه شبکه عصبی برای رگرسیون است، با این تفاوت که هر y_i یک بردار پایهٔ است که اندیس jام آن، که بیانگر طبقه داده iام است برابر با ۱ و دیگر اندیس‌های آن برابر با ۰ است و $j = 1, \ldots, k$. همچنین $y_i \in \mathbb{R}^k$ که نشان می‌دهد k طبقه داریم. همچنین لایه‌ی خروجی دارای k نورون یعنی برابر با تعداد طبقه‌ها است. همچنین در این‌حالت، خروجی هر نورون در آخرین لایه به‌صورت زیر محاسبه می‌شود:

$$z_i^{(h+1)} = softmax(h_i^{(h+1)}) = \frac{e^{h_i^{(h+1)}}}{\sum_{j=1}^{k} e^{h_j^{(h+1)}}}$$

http://dx.doi.org/10.22108/msci.2024.141722.1668

رابطه‌ی بالا بدان معناست که برای هر داده، احتمالی به طبقه iام اختصاص داده می‌شود. علاوه بر این، در تابع درست‌نمایی $p(y_j|X_j,\theta)=\prod_{i=1}^{k}\left(z_i^{(h+1)}\right)^{y_{ji}}$، در صورتی‌که خروجی توسط شبکه عصبی، با برچسب داده یکسان باشد، این تابع بیش‌ترین مقدار را می‌گیرد.

ساختار استنباط بیزی در این‌حالت مشابه شبکه عصبی بیزی برای مسئله رگرسیون است، با این تفاوت که پارامتر σ^2 در این مدل وجود ندارد. الگوریتم متروپولیس–هستینگز برای مسئله طبقه‌بندی به صورت زیر است:

الگوریتم ۳ متروپولیس–هستینگز برای مسئله طبقه‌بندی

ورودی: .

خروجی: .

شروع

۱: با مقادیر اولیه θ_0 شروع کنید.

۲: قرار دهید $t=1$.

۳: قرار دهید $\theta_t=\theta_{(t-1)}$.

۴: نمونه پیشنهادی را از توزیع $\theta_p\sim N_{|\theta|}(\theta_t,c_s^2 I_{|\theta|})$ تولید کنید.

۵: قرار دهید $\alpha=\min\left(1,\frac{p(y|x,\theta_p,\sigma_p^2)\cdot p(\theta_p)}{p(y|x,\theta_t,\sigma_t^2)\cdot p(\theta_t)}\right)$.

۶: با احتمال α قرار بدهید $\theta_t=\theta_p$ و با احتمال $1-\alpha$ قرار بدهید $\theta_t=\theta_{(t-1)}$.

۷: قرار دهید $t=t+1$.

۸: تا زمانی که $t\leq T$ بروید به مرحله ۳ وگرنه به مرحله ۹ بروید.

۹: پایان.

۴.۷. توزیع پیش‌بین پسین در مسئله طبقه‌بندی. در این‌حالت نمونه‌گیری از رابطه را به‌صورت زیر انجام می‌دهیم:

$$\hat{y}^{new}=\sum_{i=1}^{k}NN(\theta_i,x^{new})$$

که θ_i نمونه iام و $NN(\theta_i,x^{\text{new}})$ خروجی یک شبکه عصبی با پارامترهای θ_i و ورودی x^{new} را نشان می‌دهد.

۸. مقایسه عددی شبکه‌های عصبی بیزی و عادی در تحلیل داده‌ها

در این بخش، نتایج به‌کارگیری شبکه‌های عصبی عادی و شبکه‌های عصبی بیزی را در دو مسئله رگرسیون و طبقه‌بندی بررسی و مقایسه می‌کنیم.

۱.۸. مسئله رگرسیون برای مجموعه داده ریبوفلاوین. داده‌های ژنتیکی، به‌طور معمول داده‌های با ویژگی‌های زیاد و اندازه نمونه کم هستند که تحلیل رگرسیون کلاسیک در آن‌ها دقت پایینی دارد. در چنین داده‌هایی شبکه عصبی کلاسیک نمی‌تواند به خوبی متغیر پاسخ را پیش‌بینی کند. مجموعه داده ریبوفلاوین[۳۲] شامل ۴۰۸۸ متغیر تبیینی برای ژن یک گونه باسیل و ۷۱ نمونه است. همچنین متغیر پاسخ میزان تولید آنزیم ریبوفلاوین توسط باسیل سوبتیلیس[۳۳] است. در این مسئله هدف پیش‌بینی مقدار ریبوفلاوین براساس ۴۰۸۸ متغیر مستقل است. مجموعه داده را به‌صورت تصادفی به ۵۶ نمونه آموزشی

[32] riboflavin [33] Bacilluss subtilis

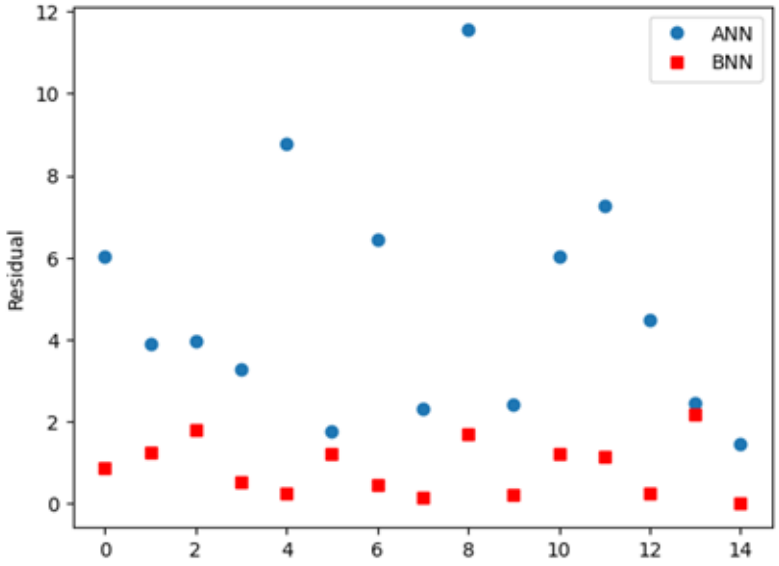
http://dx.doi.org/10.22108/msci.2024.141722.1668

(۸۰٪) و ۱۵ نمونه آزمایشی تقسیم و میزان خطا را برای داده‌های آزمایش به‌دست آورده‌ایم. نتایج جدول ۱ نشان می‌دهد شبکه‌های عصبی بیزی عملکرد بسیار بهتری نسبت به شبکه‌های عصبی عادی دارند. همچنین شکل ۵ مقدار مانده‌ها[۳۴] در این دو مدل را برای داده‌های آزمایش نشان می‌دهد. این نمودار بیان‌گر این است که شبکه عصبی بیزی، مقدار مانده‌های کمتری نسبت به شبکه عصبی عادی دارد زیرا اختلاف خروجی‌های پیش‌بینی شده و خروجی‌های واقعی در شبکه عصبی بیزی، بسیار کمتر از شبکه عصبی عادی است.

جدول ۱. مقایسه عملکرد شبکه‌های عصبی بیزی و شبکه‌های عصبی عادی در مجموعه داده ریبوفلاوین

Table 1: Comparison of the performance of BNN and standard NN on the Ribo avin dataset.

خطا	ANN	BNN
میانگین توان دوم خطا	۳۰/۷۵	۱/۲۱
میانگین قدرمطلق خطا	۴/۸۰	۰/۸۸



شکل ۵. مقدار مانده‌ها برای داده‌های ۰ تا ۱۴ در شبکه عصبی بیزی (نقاط مربع) و شبکه عصبی عادی (نقاط دایره) در مسئله رگرسیون در مجموعه داده ریبوفلاوین

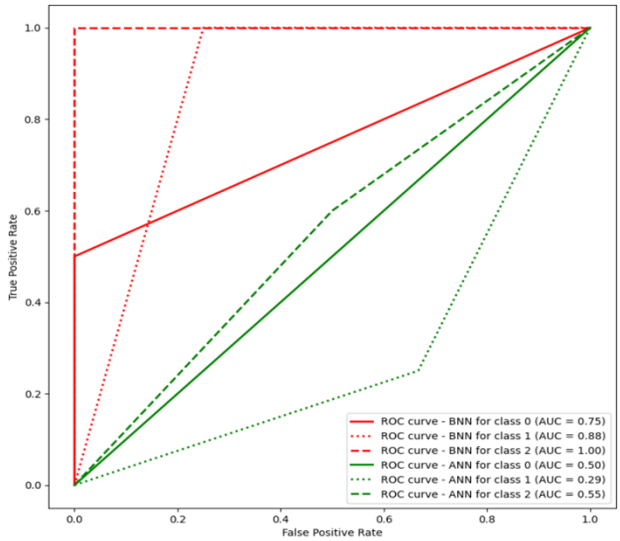
Figure 5: Residual values for 14 data points in the regression task on the Ribo avin dataset, comparing BNN (represented by square markers) and standard NN (represented by circle markers)

[34] residuals

http://dx.doi.org/10.22108/msci.2024.141722.1668

۲.۸. مسئله طبقه‌بندی چند کلاسه برای مجموعه داده سرطان ریه. اگرچه برای بسیاری از مجموعه‌های داده، شبکه عصبی بیزی با تنظیم و تعداد نمونه مناسب عملکرد بهتری نسبت به شبکه عصبی کلاسیک دارد، با این حال در این بخش مجموعه داده‌ای را انتخاب کرده‌ایم که این تفاوت را به شکل بارزی نشان دهد. برای این کار مجموعه داده‌ای با اندازه نمونه کم و تعداد ویژگی‌های زیاد انتخاب کرده‌ایم. مجموعه داده *lung cancer*، با هدف تشخیص سه نوع سرطان ریه جمع‌آوری شده و شامل ۵۶ متغیر مستقل از نوع کیفی[35] و یک متغیر خروجی با ۳ حالت که نشان‌دهنده طبقه مربوط به هر داده است می‌باشد ([۹]، وب‌سایت UCI]). تعداد نمونه‌ها ۳۲ عدد هستند. بهترین دقت‌های به دست آمده با روش تحلیل ممیزی منظم شده[36] برای این مجموعه داده ۶۲/۵ درصد گزارش شده است (https://data.world/uci/lung-cancer). براساس نتایج ۱۰ اجرا بر روی داده‌های آزمایشی که ۲۰ درصد کل نمونه‌ها هستند، شبکه عصبی بیزی با میانگین دقت ۷۱ درصد و شبکه عصبی عادی با میانگین دقت ۳۰ درصد، این داده‌ها را طبقه‌بندی می‌کند. در شکل ۶، منحنی مشخصه عامل گیرنده یا به اختصار ROC برای هر طبقه از هر طبقه‌بند رسم شده است. این نمودار نشان می‌دهد که روش شبکه عصبی کلاسیک با روش تخصیص تصادفی داده‌ها به طبقه‌ها معادل است زیرا مشاهده می‌شود که AUC نزدیک به ۰/۵ است، در حالی که شبکه عصبی بیزی بسیار خوب طبقه‌ها را تشخیص داده زیرا AUC بسیار بیشتر از ۰/۵ است.

همچنین جدول ۲ مقادیر معیارهای مختلف برای ارزیابی این دو مدل را نشان می‌دهد. همان‌طور که مشاهده می‌شود، شبکه عصبی بیزی، برحسب معیارهای مختلف ارزیابی طبقه‌بند، بسیار بهتر از شبکه عصبی عادی نتیجه داده است.



شکل ۶. منحنی ROC شبکه‌های عصبی بیزی در مقابل شبکه‌های عصبی عادی در مسئله طبقه‌بندی مجموعه داده سرطان ریه (منحنی پیوسته: برای کلاس صفر، منحنی نقطه‌چین: برای کلاس اول، منحنی خط‌چین: برای کلاس دوم)

Figure 6: ROC curve of BNN versus standard NN in classi
cation task on lung cancer dataset (The continuous curve: for class 0, dotted curve: for class 1, dashed curve: for class 2)

[35]categorical [36]regularized discriminant analysis (RDA)

http://dx.doi.org/10.22108/msci.2024.141722.1668

جدول ۲. ارزیابی شبکه‌های عصبی بیزی و شبکه‌های عصبی عادی در مجموعه داده cancer، lung برحسب چهار معیار رایج

Table 2: Evaluation of BNN and NN on the lung cancer dataset based on four common metrics.

معیار	ANN	BNN
دقت	۰/۴۲	۰/۸۵
اندازه–F	۰/۳۸	۰/۸۴
نمره $F-Beta$	۰/۳۶	۰/۸۶
نمره یاکارد	۰/۲۷	۰/۷۵

۹. جمع‌بندی و نتیجه‌گیری

شبکه عصبی بیزی به‌عنوان یک جایگزین کارآمد برای شبکه‌های عصبی کلاسیک به‌خصوص در شرایطی که نمونه کافی برای برآورد پارامترهای شبکه عصبی وجود ندارد معرفی و مرور شد. دست‌کم در دو مثال کاربردی ملاحظه شد که در چنین شرایطی شبکه عصبی بیزی با اختلاف قابل توجهی بهتر از شبکه عصبی کلاسیک عمل می‌کند. از آن‌جا که توزیع پیشین و پسین مزدوج برای شبکه‌های عصبی با توجه به توابع فعال‌ساز مورد استفاده وجود ندارد، به‌منظور تقریب توزیع پسین، روش نمونه‌گیری متروپولیس–هستینگز را استفاده کردیم. با این‌حال این روش، درحالت بعد بالای پارامترها، هزینه محاسباتی بالایی دارد. بنابراین، در پژوهش‌های اخیر به استفاده از روش‌های تقریب توزیع پسین، کم‌هزینه‌تر مانند بیز تغییراتی[۳۷] روی آورده شده است. از جمله این پژوهش‌ها می‌توان به [۳]، [۱۴]، [۱۷] و [۱۸] اشاره کرد. در کارهای آینده می‌توان از روش‌های تقریب سریع‌تر مانند بیز تغییراتی و بهبود روش‌های نمونه‌گیری مانند MCMC برای شبکه‌های عصبی بیزی استفاده کرد.

مراجع

[۱] س. م. طاهری، آمار و شبکه‌های عصبی مصنوعی، مجموعه مقالات هشتمین کنفرانس آمار ایران، دانشگاه شیراز، (۱۳۸۵) ۸۱–۹۱.

[۲] م. ر. مشکانی و ا. کاوسی دولانقر، روش‌های آمار بیزی، انتشارات دانشگاه علوم پزشکی دانشگاه شهید بهشتی، ۱۴۰۱.

[3] C. Blundell, J. Cornebise, K. Kavukcuoglu and D. Wierstra, Weight uncertainty in neural network, *Proc. of the International Conference on Machine Learning PMLR*, Lille, France, (2015) 1613–1622.

[4] J. P. Bharadiya, A review of Bayesian machine learning principles, methods, and applications, *Int. J. Innov. Sci. Res. Technol.*, **8** no. 5 (2023) 2033–2038.

[5] R. Chandra, R. Chen and J. Simmons, Bayesian neural networks via MCMC: a Python-based tutorial, (2023). https://doi.org/10.48550/arXiv.2304.02595.

[6] C. M. Carlo, Markov chain monte carlo and gibbs sampling, *Lecture Notes for EEB 581*, (2004) 24 p.

[7] A. Graves, Practical variational inference for neural networks, Part of *Part of Advances in Neural Information Processing Systems 24 (NIPS)*, (2011)

[8] A. Gelman, J. B. Carlin, H. S. Stern and D. B. Rubin, *Bayesian Data Analysis*, Chapman and Hall/CRC, 1995.

[37] variational bayes

[9] Z. Q. Hong and J. Y. Yang, Lung cancer, UCI Machine Learning Repository, (1992). https://doi.org/10.24432/C57596.

[10] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, **57** no. 1 (1970) 97–109.

[11] L. V. Jospin, H. Laga, F. Boussaid, W. Buntine and M. Bennamoun, Hands-on Bayesian neural networks—A tutorial for deep learning users, *IEEE Computational Intelligence Magazine*, **17** no. 2 (2022) 29–48.

[12] H. D. Kabir, A. Khosravi, M. A. Hosen and S. Nahavandi, Neural network-based uncertainty quantification: A survey of methodologies and applications, *IEEE Access*, **6** (2018) 36218–36234.

[13] J. Ker, L. Wang, J. Rao and T. Lim, Deep learning applications in medical image analysis, *IEEE Access*, **6** (2018) 9375–9389.

[14] I. Oleksiienko, D. T. Tran and A. Iosifidis, Variational neural networks, *Procedia Computer Science*, **222** (2023) 104–113.

[15] C. P. Robert, G. Casella and G. Casella, *Monte carlo statistical methods*, **2**, Springer, 1999.

[16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus, Intriguing properties of neural networks, (2013). arXiv preprint arXiv:1312.6199. https://doi.org/10.48550/arXiv.1312.6199

[17] S. Sun, G. Zhang, J. Shi and R. Grosse, Functional variational Bayesian neural networks, (2019). arXiv preprint arXiv:1903.05779. https://doi.org/10.48550/arXiv.1903.05779.

[18] M. N. Tran, T. N. Nguyen, and V. H. Dao, A practical tutorial on variational Bayes, (2021) 43 p. arXiv preprint arXiv:2103.01327. https://doi.org/10.48550/arXiv.2103.01327

[19] M. J. Zaki and W. Meira Jr, *Data mining and machine learning: fundamental concepts and algorithms*, Cambridge University Press, 2020.

معین منعمی
گروه الگوریتم‌ها و محاسبات، دانشکده علوم مهندسی، دانشکدگان فنی، دانشگاه تهران
moein.monemi@ut.ac.ir

معین منعمی متولد تیر ماه ۱۳۷۹ در شهر مشهد است. مدرک کاردانی و کارشناسی خود را در رشته مهندسی برق (گرایش الکترونیک) به ترتیب در سال‌های ۱۳۹۹ و ۱۴۰۱ از دانشگاه فنی‌وحرفه‌ای شهید منتظری مشهد دریافت نمود. در مهر ماه سال ۱۴۰۱ در مقطع کارشناسی ارشد رشته مهندسی کامپیوتر (گرایش الگوریتم‌ها و محاسبات) در دانشگاه تهران پذیرفته شد و هم‌اکنون دانشجوی کارشناسی ارشد در این رشته است.

http://dx.doi.org/10.22108/msci.2024.141722.1668

سید محمود طاهری
گروه الگوریتم‌ها و محاسبات، دانشکده علوم مهندسی، دانشکدگان فنی، دانشگاه تهران
sm_taheri@ut.ac.ir

سید محمود طاهری، کارشناسی و کارشناسی ارشد را در رشته آمار، به ترتیب، در سال‌های ۱۳۶۷ و ۱۳۷۰ از دانشگاه فردوسی مشهد و دکتری را در سال ۱۳۷۸ از دانشگاه شیراز، در رشته آمار (گرایش استنباط آماری) اخذ کرد. وی پس از حدود ۱۵ سال عضویت هیئت علمی در دانشگاه صنعتی اصفهان، از سال ۱۳۹۱ به دانشکده فنی (اکنون با تغییر نام: دانشکدگان فنی) دانشگاه تهران منتقل شد و در زمینه‌های استنباط آماری، آمار و احتمال فازی، مدل‌سازی رگرسیونی، آموزش مهندسی و محاسبات نرم فعالیت‌های آموزشی و پژوهشی دارد.

سید مرتضی امینی
بخش آمار، دانشکده ریاضی آمار و علوم کامپیوتر دانشکدگان علوم، دانشگاه تهران
morteza.amini@ut.ac.ir

سید مرتضی امینی متولد شهریور ماه ۱۳۶۰ در شهر مشهد است. وی در سال ۱۳۷۸ وارد مقطع کارشناسی رشته آمار دانشگاه شهید باهنر کرمان شد و در سال ۱۳۸۲ مدرک کارشناسی آمار خود را دریافت نمود. سپس در فاصله سال‌های ۱۳۸۳ تا ۱۳۸۶ به تحصیل در دوره کارشناسی ارشد آمارریاضی در دانشگاه فردوسی مشهد پرداخت. او در سال ۱۳۸۶ در دوره دکتری تخصصی آمار در دانشگاه فردوسی مشهد پذیرفته شد و در سال ۱۳۹۰ توانست مدرک دکتری خود را دریافت نماید. از سال ۱۳۹۰ در دانشگاه تهران به‌عنوان عضو هیات علمی مشغول به کار شد. در حال حاضر او دانشیار آمار دانشگاه تهران است.